AMENDMENT TO THE CLAIMS

CLAIM 1 (canceled)

CLAIM 2 (canceled)

CLAIM 3 (canceled)

CLAIM 4 (canceled)

CLAIM 5 (original):

CLAIM 6 (canceled)

CLAIM 7 (canceled)

CLAIM 8 (currently amended):

8. ~~The method of claim 1~~ A method of allocating and deallocating memory comprising the steps of:

assigning to each basic unit of user memory a corresponding memory control block;

collecting groups of contiguous available control blocks into a plurality of linked lists, each list for storing a plurality of elements, each element comprising a control block group having an associated minimum size user block;

in response to a request for a block of user memory, searching for a linked list having available control block groups corresponding to user blocks at least as large as the requested size;

seizing a block of user memory of the required size and making available any surplus block representing a difference between the requested size of memory and the size of the seized block of user memory;

when deallocating memory, testing whether user blocks of memory immediately adjacent to the deallocated block are available and if available merging the available blocks to the block being deallocated to create a merged deallocated block; and

inserting the merged deallocated block into a linked list of available blocks of memory for containing blocks of memory of at least the size of the merged block;

whereby the adding of said surplus block and the process of creating a merged deallocated block helps to avoid fragmentation of memory;

further comprising the ~~step~~ steps of:

~~storing availability bits for each basic unit of user memory;~~

storing <u>a</u> <u>separate</u> <u>busy</u> <u>bit</u> <u>map</u> <u>and</u> <u>a</u> <u>separate</u> <u>idle</u> <u>bit</u> <u>map,</u> <u>each</u> <u>bit</u> <u>of</u> <u>each</u> <u>map</u> <u>having</u> <u>a</u> <u>one-to-one</u> <u>correspondence</u> <u>with</u> <u>the</u> <u>control</u> <u>blocks;</u>

<u>wherein</u> <u>each</u> <u>active</u> <u>busy</u> <u>bit</u> <u>represents</u> <u>a</u> <u>beginning</u> <u>of</u> <u>a</u> <u>control</u> <u>block</u> <u>group</u> <u>for</u> <u>active</u> <u>user</u> <u>memory;</u>

<u>wherein</u> <u>each</u> <u>active</u> <u>idle</u> <u>bit</u> <u>represents</u> <u>the</u> <u>beginning</u> <u>of</u> <u>a</u> <u>control</u> <u>block</u> <u>group</u> <u>for</u> <u>available</u> <u>user</u> <u>memory;</u>

in case said memory control blocks are inadvertently overwritten, recreating a new ~~set of~~ linked lists <u>of</u> <u>available</u> <u>memory</u> from data of said ~~availability bits~~ <u>busy</u> <u>bit</u> <u>map</u> <u>and</u> <u>said</u> <u>idle</u> <u>bit</u> <u>map.</u>

CLAIM 9 (canceled)

CLAIM 10 (canceled)

CLAIM 11 (canceled)

CLAIM 12 (canceled)

CLAIM 13 (canceled)

CLAIM 14 (canceled)

CLAIM 15 (canceled)

CLAIM 16 (canceled

CLAIM 17 (canceled)

CLAIM 18 (currently amended)

18. ~~The memory arrangement of claim 11~~ <u>A</u> memory arrangement <u>comprising:</u>

<u>a</u> <u>plurality</u> <u>of</u> <u>contiguous</u> <u>basic</u> <u>units</u> <u>of</u> <u>user</u> <u>memory;</u>

<u>a</u> <u>corresponding</u> <u>memory</u> <u>control</u> <u>block</u> <u>assigned</u> <u>to</u> <u>each</u> <u>basic</u> <u>unit</u> <u>of</u> <u>memory;</u>

<u>groups</u> <u>of</u> <u>contiguous</u> <u>available</u> <u>control</u> <u>blocks</u> <u>are</u> <u>collected</u> <u>into</u> <u>a</u> <u>plurality</u> <u>of</u> <u>linked</u> <u>lists,</u> <u>each</u> <u>list</u> <u>for</u> <u>storing</u> <u>a</u> <u>plurality</u> <u>of</u> <u>elements,</u> <u>each</u> <u>element</u> <u>comprising</u> <u>a</u> <u>control</u> <u>block</u> <u>group</u> <u>having</u> <u>an</u> <u>associated</u> <u>minimum</u> <u>size</u> <u>user</u> <u>block;</u>

<u>for</u> <u>responding</u> <u>to</u> <u>a</u> <u>request</u> <u>for</u> <u>a</u> <u>block</u> <u>of</u> <u>user</u> <u>memory,</u> <u>an</u> <u>element</u> <u>of</u> <u>a</u> <u>linked</u> <u>list</u> <u>having</u> <u>an</u> <u>available</u> <u>block</u> <u>of</u> <u>user</u> <u>memory</u> <u>at</u> <u>least</u> <u>as</u> <u>large</u> <u>as</u> <u>a</u> <u>requested</u> <u>size</u> <u>of</u> <u>an</u> <u>allocation</u> <u>request;</u>

when seizing a block of user memory of the required size indicated by said linked list, making available any surplus representing a difference between a requested allocation size, and a size of the seized block of user memory;

when deallocating memory, testing whether user blocks of memory immediately adjacent to a deallocated block are available and if available merging the available blocks to the block being deallocated to create a merged deallocated block; and

inserting the merged deallocated block into a linked list of available contiguous control blocks for blocks of user memory of at least the size of the merged deallocated block;

further comprising:

~~availability bits for each basic unit of user memory;~~

means for storing a separate busy bit map and a separate idle bit map, each bit of each map having a one-to-one correspondence with the control blocks;

wherein each active busy bit represents a beginning of a control block group for active use memory;

wherein each active idle bit represents the beginning of a control block group for available user memory;

in case said memory control blocks are inadvertently overwritten, a new set of linked lists can be created from data of said availability bits.

CLAIM 19 (canceled)

CLAIM 20 (canceled)

CLAIM 21 (new)]

21. A method of allocating and deallocating user memory comprising the steps of:

creating linked lists of groups of continuous groups of basic units of available user memory;

storing a separate busy bit map and a separate idle bit map, each bit of each map having a one-to-one correspondence with one basic unit of user memory;

wherein each active busy bit represents a beginning of a group of basic units of active user memory;

wherein each active idle bit represents the beginning of a continuous group of basic units of available user memory; and

wherein in case portions of said user memory are inadvertently overwritten, recreating a new set of linked lists of available user memory from data of said busy bit map and said idle bit map.

CLAIM 22 (new)

22. A user memory arrangement comprising:

a plurality of contiguous basic units of user memory;

linked lists of groups of continuous groups of basic units of available user memory;

means for storing a separate busy bit map and a separate idle bit map, each bit of each map having a one-to-one correspondence with one basic unit of user memory;

wherein each active busy bit represents a beginning of a group of active basic units of user memory;

wherein each active idle bit represents the beginning of a group of available basic units of user memory; and

wherein, in case portions of said user memory are inadvertently overwritten, new linked lists can be created from data of said availability bits.